

What is Claimed is:

- [c1] A method for optimizing data searches in tree structures, the method comprising:
- organizing multiple search levels of data into sub-trees contained in fixed size blocks of shared external memory of an embedded processing system; and
 - requiring each reference to the data to proceed from one-half of a sub-tree during a descent of the search tree based on a search pattern.
- [c2] The method of claim 1 further comprising choosing the one-half before the descent from a root level of the search tree to reduce the size of the required reference.
- [c3] The method of claim 2 wherein the step of choosing further comprises choosing the one-half based on a next bit test value of a bit in the search pattern.
- [c4] The method of claim 3 wherein the one-half further comprises a right side or left side of a branch table of the sub-tree.
- [c5] The method of claim 1 wherein organizing multiple search levels further comprises organizing the sub-trees to provide multiple levels of data with each reference.
- [c6] A system for optimizing data searches in tree structures, the system comprising:
- an embedded processor, the embedded processor including a tree search engine; and
 - external memory coupled to the embedded processor, the external memory containing multiple search levels of data as sub-trees in fixed size blocks and providing data from one-half of a sub-tree during a search tree descent by the tree search engine based on a search pattern.
- [c7] The system of claim 6 wherein the tree search engine chooses the one-half before the descent from a root level of the search tree to reduce the size of a required reference.

- [c8] The system of claim 7 wherein the tree search engine chooses the one-half based on a next bit test value of a bit in the search pattern.
- [c9] The system of claim 8 wherein the one-half further comprises a right side or left side of a branch table of the sub-tree.
- [c10] The system of claim 6 wherein the multiple search levels further comprise organized sub-trees that provide multiple of data with each reference.
- [c11] A method for optimizing data searches in tree structures, the method comprising:
- utilizing external memory in an embedded processing system;
 - organizing the external memory in multiple search levels of data as sub-trees in fixed size blocks; and
 - providing data from one-half of a sub-tree during a search tree descent by a tree search engine based on a search pattern.
- [c12] The method of claim 11 further comprising choosing the one-half before the descent from a root level of the search tree to reduce the size of a required reference.
- [c13] The method of claim 12 wherein the step of choosing further comprises choosing the one-half based on a next bit test value of a bit in the search pattern.
- [c14] The method of claim 13 wherein the one-half further comprises a right side or left side of a branch table of the sub-tree.
- [c15] The method of claim 11 wherein organizing multiple search levels further comprises organizing the sub-trees to provide multiple levels of data with each reference.
- [c16] A computer readable medium containing program instructions for optimizing data searches in tree structures, the program instructions comprising:
- organizing multiple search levels of data into sub-trees contained in fixed size blocks of shared external memory of an embedded processing system; and

requiring each reference to the data to proceed from one-half of a sub-tree during a descent of the search tree based on a search pattern.

- [c17] A computer readable medium containing program instructions for optimizing data searches in tree structures, the program instructions comprising:
- utilizing external memory in an embedded processing system;
 - organizing the external memory in multiple search levels of data as sub-trees in fixed size blocks; and
 - providing data from one-half of a sub-tree during a search tree descent by a tree search engine based on a search pattern.